Mathematical Biosciences and Engineering

MBE, 16(6): 6858–6873.
DOI: 10.3934/mbe.2019343
Received: 14 May 2019
Accepted: 23 July 2019
Published: 29 July 2019

http://www.aimspress.com/journal/MBE

Research article

Ensemble learning of model hyperparameters and spatiotemporal data for calibration of low-cost $PM_{2.5}$ sensors

Peng-Yeng Yin[1,2,*], Chih-Chun Tsai[1], Rong-Fuh Day[1], Ching-Ying Tung[2] and Bir Bhanu[3]

[1] Department of Information Management, National Chi Nan University, Nantou, 54561, Taiwan, ROC
[2] Institute of Strategy and Development of Emerging Industry, National Chi Nan University, Nantou, 54561, Taiwan, ROC
[3] Center for Research in Intelligent Systems, University of California, Riverside, California 92521, USA

*** Correspondence:** Email: pyyin@ncnu.edu.tw; Tel: +886492910960; Fax: +886492915205.

Abstract: The $PM_{2.5}$ air quality index (AQI) measurements from government-built supersites are accurate but cannot provide a dense coverage of monitoring areas. Low-cost $PM_{2.5}$ sensors can be used to deploy a fine-grained internet-of-things (IoT) as a complement to government facilities. Calibration of low-cost sensors by reference to high-accuracy supersites is thus essential. Moreover, the imputation for missing-value in training data may affect the calibration result, the best performance of calibration model requires hyperparameter optimization, and the affecting factors of $PM_{2.5}$ concentrations such as climate, geographical landscapes and anthropogenic activities are uncertain in spatial and temporal dimensions. In this paper, an ensemble learning for imputation method selection, calibration model hyperparameterization, and spatiotemporal training data composition is proposed. Three government supersites are chosen in central Taiwan for the deployment of low-cost sensors and hourly $PM_{2.5}$ measurements are collected for 60 days for conducting experiments. Three optimizers, Sobol sequence, Nelder and Meads, and particle swarm optimization (PSO), are compared for evaluating their performances with various versions of ensembles. The best calibration results are obtained by using PSO, and the improvement ratios with respect to R^2, RMSE, and NME, are 4.92%, 52.96%, and 56.85%, respectively.

Keywords: ensemble learning; low-cost sensors; air quality index; particle swarm optimization; $PM_{2.5}$; spatiotemporal data; sensor calibration

1. Introduction

The immense amount of industry productions and anthropogenic activities exasperate the concentrations of particulate matter with aerodynamic diameter ≤ 2.5 μm ($PM_{2.5}$) in natural environment. Many researches have generated evidence for the strong correlation between ambient $PM_{2.5}$ concentrations and human health [1], climate change [2], atmospheric visibility [3], plant species mortality [4], to name a few. The transportation and dispersion path of $PM_{2.5}$ is hard to analyze and predict due to many uncertain anthropogenic activities (such as vehicle exhaust, coal and gasoline combustion, petrochemical production, and steel refinery), and mother-nature scenarios (such as soils, crustal elements, volcanic eruptions, wind and precipitation, typhoons, and landscapes). These uncertain factors span in both spatial and temporal dimensions. To estimate the actual $PM_{2.5}$ concentrations, expensive and sparsely-distributed supersite sensors have been built by the government to monitor possible contaminations at a few regions of interest.

As the $PM_{2.5}$ supersites are costly, they are sparsely installed in the monitoring area, lacking the ability to provide a satisfactory coverage of the investigated field. Thus, establishing internet of things (IoT) with low-cost and low-power sensors is emerging as a complement to the supersites and has been implemented in several countries. Hu et al. [5] constructed a sensor network named HazeEst which used machine learning techniques to estimate air pollution surface in Sydney by combining data from government-built fixed supersites and personally-affordable mobile sensors. Miksys [6] deployed inexpensive $PM_{2.5}$ and PM_{10} sensors to conduct spatiotemporal predictions of fine-grained resolutions in Edinburgh. Chen et al. [7] deployed a participatory urban sensing network for $PM_{2.5}$ monitoring with more than 2500 sensors in Taiwan and 29 other countries.

Although low-cost sensors can provide denser monitoring networks than those offered by supersites, the obtained $PM_{2.5}$ measurements are less accurate. A feasible solution is to calibrate low-cost sensors by finding the relationship function between the measurements of low-cost sensors and supersite sensors. The relationship function can be found by mathematical regression, support vector regression, gradient regression tree boosting, adaptive neuro-fuzzy inference system (ANFIS), to name a few [8]. Many researches have also shown the geographical landscapes and meteorological patterns that have various degrees of influence on $PM_{2.5}$ concentrations [9–11], and this influence deteriorates along the spatial and temporal distances. We contemplate the exploitation of spatiotemporal data and the model for data imputation and calibration can improve the measurement accuracy of low-cost sensors.

The contributions of this paper include the following. (1) The deployment of low-cost $PM_{2.5}$ sensors that provides a denser coverage of air-quality monitoring area than that with government-built supersites. Our ensemble for imputation and calibration learning enhances the accuracy of measured air quality index (AQI) by low-cost $PM_{2.5}$ sensors. (2) The dynamics of $PM_{2.5}$ concentrations depend on spatial and temporal factors. We include the spatiotemporal learning by finding the best composition of training data in both spatial and temporal dimensions. (3) We develop an ensemble method from a holistic point of view where the best selection strategy for imputation method, the hyperparameter values of calibration model, and the composition of spatiotemporal training data, are learned by an effective optimizer. The experimental results manifest that our ensemble learning calibrates the low-cost sensors by enhancing R^2, RMSE, and NME, with a significant improvement of 4.92%, 52.96%, and 56.85%, respectively.

The remainder of this paper is organized as follows. Section 2 describes the state-of-the-art

approaches for data imputation and sensor calibration. Section 3 presents the proposed ensemble learning for best imputation method, calibration of hyperparameter values, and spatiotemporal data composition. Section 4 provides the experimental results with discussions. Finally, Section 5 concludes this paper.

2. Related work

2.1. Data imputation

Missing value is a commonly encountered problem in IoT applications. The reasons for incurring missing values could result from direct failures of sensors, linkage failures or data losses in network communication, malfunction of storage servers, or blackout of electricity. Some missing-value scenarios can be eradicated by generating multiple duplicates of data and storing them in distributed storage servers [12]. But this approach entails a large volume of storage and is not able to deal with sensor failures and electricity blackout. An alternative to the data-redundancy approach is data imputation by use of statistics or machine learning. Data imputation approaches estimate the missing values by analyzing the covariance of existing variable values or learning the multivariate relations. The data imputation approaches are useful when the data loss is random and dependency exists among variables.

The *mean imputation* method [13] is the simplest imputation method which replaces the missing values by the mean of existing data for the corresponding variable. The mean imputation method does not change the variable mean, however, some statistics such as variance and standard deviation are underestimated. The *interpolation imputation* method [14] assumes the original value of the missing data has mathematical relations with its neighboring data of the same variable and thus can be restored by various forms of interpolation, such as linear, triangular, weighted, or higher-order forms. The *KNN imputation* method [15] discards the variable of the missing value and uses the remaining variable values to search the k nearest records in appropriate distance space. The missing value is then filled up by the distance-weighted mean of the values existing in its neighbors. The *MICE imputation* method [16] is a regression method where the imputed value is predicted from a regression equation. The *SOFT imputation* method [17] treats the imputation as a matrix completion problem and uses the convex relaxation technique to provide a sequence of regularized low-rank solutions and iteratively replace the missing values with those obtained from a soft-thresholded singular value decomposition.

2.2. Sensor calibration

The low-cost sensors are deployed with a dense coverage than the government-built expensive sensors. However, the readings obtained from low-cost sensors are not highly accurate and should be carefully adjusted before being released for applications. Sensor calibration is such a process by either hardware adjustment or software manipulation. As our sensors are low-cost and hardware calibration is not feasible, we focus on software calibration in this paper. The government-built high-accuracy sensors provide good references for software calibration of low-cost sensors. When a low-cost sensor and a high-accuracy sensor is near enough, they should read the same measured value. Therefore, the software calibration can be fulfilled by mathematical equations such as

regression. In fact, the US EPA and Taiwan EPA apply regression technique to calibrate sensors by reference to manual measurements. Every year, Taiwan EPA releases the linear regression equations adopted by automatic sensors (https://taqm.epa.gov.tw/pm25/tw/Download/). As machine learning approaches, such as support vector regression [18] and gradient boosted regression tree [19], usually manifest better performance over mathematical regressions, this paper adopts XGBoost [20], which is one of the best machine learning regression methods, as our calibration model.

XGBoost is a novel gradient tree boosting algorithm which has won several competitions including Kaggle's challenges (https://www.kaggle.com/competitions) and KDDCup 2015 [21]. By using a sparsity-aware split-finding algorithm and weighted quantile sketch, XGBoost machine scales up to billions of data examples but only consume fewer computational resources than other machine learning regression methods. XGBoost machine has a number of hyperparameters which influence the learned ensemble of regression trees between the observations and variable values. The hyperparameters and their corresponding value ranges are described in Table 1.

Table 1. Value ranges and connotations of XGBoost hyperparameters.

Parameters	Ranges	Connotations
g_1	[1, 4]	Tree maximal level
g_2	[1, 300]	Number of boosting trees
g_3	[0, 12]	Minimum weighted sum of leaf nodes
g_4	[0.001, 0.9]	Learning rate
g_5	[0, 1.0]	Proportion of training data
g_6	[0, 2.0]	Threshold for split finding
g_7	[0, 2.0]	L1 regularization term
g_8	[0, 2.0]	L2 regularization term

The hyperparameter optimization of a machine learning algorithm is a process which searches the most appropriate setting values of the hyperparameters to obtain the optimal performance of the machine learning algorithm on the addressed problem. The hyperparameter optimization can be achieved by random search [22], Taguchi orthogonal method [23], and meta-learning. The meta-learning method learns the optimal configuration of machine learning hyperparameters by another machine learning approach. This fashion of learning-for-learning has attracted many researchers [18,24].

In addition to the hyperparameter optimization of the adopted machine learning approach, the selection of the most appropriate training instances is very critical [25]. The instance selection technique not only reduces the data volume but also increases the accuracy of the discovered knowledge from a big dataset. For learning the $PM_{2.5}$ concentration dataset, the selection of span coverage of spatial and temporal training instances is related to the local landscape, climate, and land usage. Hence, the conception of learning on selection of spatiotemporal $PM_{2.5}$ training instances is particularly useful and it has not been explored in the related literature.

3. Method

3.1. Ensemble learning

We propose an ensemble method to learn the best configuration of three cooperating tasks: the selection of data imputation methods, the hyperparameter optimization of XGBoost model for data calibration, and the selection of spatiotemporal training instances. Our system concept is illustrated in Figure 1. The available imputation methods we considered are the mean method, interpolation method, KNN, MICE, and SOFT. The hyperparameters of XGBoost need to be tuned are the eight parameters as described in Table 1. There are two choices of spatial training data for each low-cost sensor: using the sequence of readings of its own and its referred supersite, or using the sequence of readings of all supersites and low-cost sensors. The available temporal training data is 30-day historical hourly $PM_{2.5}$ measurements. We, thus, consider temporal data in t days for calibration where t is selected between 1 and 30.

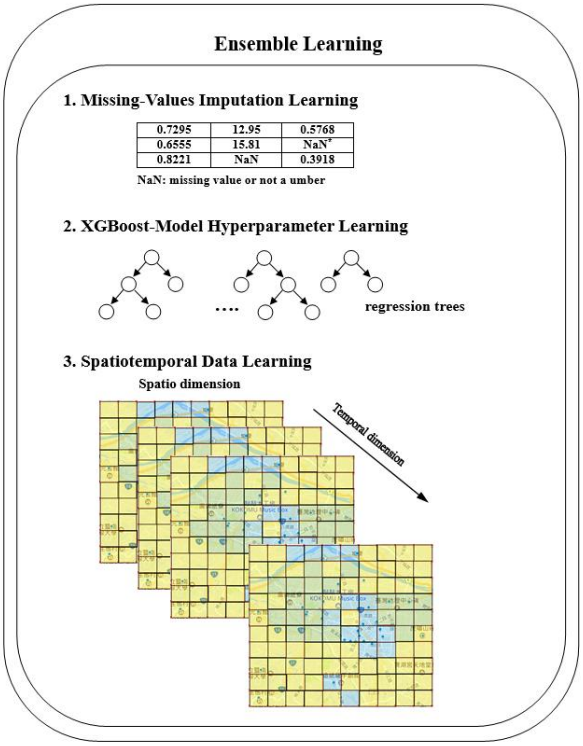


Figure 1. Concept diagram of the proposed approach.

I	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	C	t

Figure 2. Representation of the ensemble consisting of imputation method, XGBoost hyperparameters, spatial data type, and temporal data range.

To construct an ensemble learning of imputation, XGBoost hyperparameters, spatial data collection, and temporal data ranges, the task is formulated as a decision problem involving 11 continuous or integer variables. The decision ensemble solution is represented as a vector, as shown in Figure 2, encoding a selection instance of imputation method (*I*), XGBoost hyperparameters (g_1, g_2, …, g_8), spatial data collection (*C*), and temporal data range (*t*). To learn the best ensemble, an optimization method needs to be applied. In the experiments, we will compare the performance of three optimizers which are briefly described as follows.

3.2. Compared optimizers for ensemble learning

3.2.1. Particle swarm optimization

The particle swarm optimization (PSO) [26] is an evolutionary algorithm which is capable of learning the optimal value of model hyperparameters. PSO is a bio-inspired algorithm which mimics the social dynamics of bird flocking. This form of social intelligence not only increases the success rate for food foraging but also expedites the process. Considering a swarm of *N* particles $\{x^1, x^2, \cdots x^N\}$ in an *s*-dimensional Euclidean space, the particle moving trajectory is guided by the personal best (*pbest*) and the global best (*gbest*). Particle *i* has a personal memory storing the best position among those it has visited, referred to as *pbest*, and the best position *gbest* visited by the entire swarm. The PSO iterates a swarm evolution until a stopping criterion is satisfied, which is usually set as a maximum number of iterations. At each iteration, particle *i* adjusts its position x^i as follows.

$$v_j^i \leftarrow K\left(v_j^i + c_1 r_1 (pbest - x_j^i) + c_2 r_2 (gbest - x_j^i)\right),\ j = 1,..., s \tag{1}$$

$$x_j^i \leftarrow x_j^i + v_j^i,\ j = 1,..., s \tag{2}$$

where *K* is the constriction factor, c_1 and c_2 are the accelerating coefficients, and r_1 and r_2 are random numbers drawn from (0, 1). We designate the performance of the ensemble learning solution as the particle fitness.

3.2.2. Sobol quasirandom sequence

Sobol quasirandom sequence is a distribution of points whose function values sum up toward to the function integral, and converge as fast as possible [27]. Hence, the Sobol quasirandom sequence can be used to generate a small set of samples to reasonably well explore a large space. Considering a Sobol quasirandom sequence of *N* points $\{x^1, x^2, \cdots x^N\}$ in an *s*-dimensional Integer space, the next drawn point should minimize the inter-point discrepancy. For a point $x = (x_1, x_2, \cdots, x_s) \in Z^s$, a hypercube G_x is defined as $G_x = [0, x_1) \times [0, x_2) \times \cdots \times [0, x_s)$. The next point *x* is generated to minimize the discrepancy as follows.

$$\sup_{x \in Z^s} |S_N(G_x) - N x_1 x_2 \cdots x_s| \tag{3}$$

Due to its quasirandom and fast convergence properties, Sobol quasirandom sequence can be applied to search the near-optimal values of the ensemble learning instance.

3.2.3. Nelder and meads

Nelder and Meads (N&M) [28] is a direct search heuristic which uses a simplex of S+1 vertices $\{x^1, x^2, \cdots x^{S+1}\}$, $x^i \in R^s$, in an s-dimensional Euclidean space to conduct iterative moves towards the global optimum. Starting with an initial simplex, the N&M repeatedly replaces the worst vertex (in terms of the objective value) by an improving trial point obtained by performing reflection, expansion, or contraction operations. If no such improving trial point can be produced, the simplex shrinks its size by dragging the remaining vertices toward the best vertex and repeats the iterative moving process. Without loss of generality, let f be the objective function to be minimized and $f(x^1) \leq \cdots \leq f(x^S) \leq f(x^{S+1})$. We calculate the centroid x^o of $\{x^1, x^2, \cdots x^S\}$ and reflect x^{S+1} against x^o to obtain the reflection point x^r as follows.

$$x^r = x^o + (x^o - x^{s+1}) \tag{4}$$

If $f(x^1) \leq f(x^r) \leq f(x^n)$, then replace x^{S+1} with x^r and restart with the new simplex. If $f(x^r) < f(x^1)$, then produce the expansion point x^e as follows.

$$x^e = x^o + 2(x^o - x^{s+1}) \tag{5}$$

If $f(x^e) < f(x^r)$, then replace x^{S+1} with x^e, otherwise replace x^{S+1} with x^r. However, if $f(x^s) < f(x^r)$, then a contraction point x^c should be generated as follows.

$$x^c = x^o + 0.5(x^{s+1} - x^o) \tag{6}$$

If $f(x^c) < f(x^{s+1})$, then replace x^{S+1} with x^c, otherwise shrink the simplex by retaining x^1 and replace the remaining vertices by

$$x^i = x^1 + 0.5\left(x^i - x^1\right) \tag{7}$$

and resume the process with the new simplex until the simplex size is less than a threshold.

4. Results and discussions

We selected three government-built $PM_{2.5}$ supersites located in central Taiwan area and deployed a low-cost sensor for calibration test with each of the supersites. All the low-cost sensors are the same product model G7 PMS7003 which is a particle concentration sensor based on laser light scattering. The working principle is to use a laser to illuminate the suspended particles in the air to generate light scattering. The scattered light is then collected at a certain angle to estimate the particle size and the number of particles of different sizes per unit volume. The minimum measurable particle diameter is 0.3 μm and the sampling response time is less than one second. Taiwan EPA supersites adopt beta ray attenuation method. By gauging the beta ray decrement due to passage through a filter paper with particle concentrations, the number of particles and their sizes can be estimated. The model G7 PMS7003 low-cost sensor has been shown to have a high correlation with government-built supersite sensors in several publications [29,30]. We deployed the low-cost sensor at the places as close to as possible to the corresponding supersite. However, due to the prohibitive zone of government property, the actual distance between the corresponding low-cost sensor and supersite is between 85 and 122 meters as shown in Table 2. Fortunately, all the three supersites are

located in large open spaces and the low-cost sensors are deployed at nearby school campus, the difference between the sensor readings is mainly due to hardware characteristics and the influence of local emissions is kept minimal.

Table 2. Distance between low-cost and supersite sensors.

Low-cost sensors	Nearest supersite sensors	Distance (m)
A1	B1	122
A2	B2	85
A3	B3	112

The Taiwan EPA calibrates the automatic supersite by using a reference to the manual supersite on an annual basis. That is, the readings for the entire year from both supersites are collected for training, and the regression equation so obtained is used to calibrate the readings from the automatic supersite for the next year. As the low-cost sensors are less robust than the supersites, we chose to calibrate the low-cost sensors on a monthly basis. The monitored hourly $PM_{2.5}$ data is from the period September 24 to November 22 in 2017, spanning 60 days. The data collected in the first 30 days are used for the selection of spatiotemporal training data and the data for the remaining days are used for testing. We conduct two series of experiments. The first one is for estimating the significance of missing-value imputation and the second one is for validating the contribution of ensemble learning in a low-cost sensor calibration. The platform for conducting the experiments is a notebook computer equipped with an Intel Core i5 CPU and 8.0 GB RAM. All programs are coded in Python with machine learning open packages.

To evaluate the performance of our ensemble learning on data imputation and calibration, we adopt the following measures: the coefficient of determination (R^2), the root mean square error (RMSE), and the normalized mean error (NME). Given a set of n observed values $\{y_1, y_2, \ldots, y_n\}$ and another set of n calibrated numbers $\{x_1, x_2, \ldots, x_n\}$, we evaluate the calibration performance as follows.

$$\mathrm{R}^2 = 1 - \sum_{i=1}^{n}(y_i - x_i)^2 \Big/ \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{8}$$

$$\mathrm{RMSE} = \left(\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2\right)^{\frac{1}{2}} \tag{9}$$

$$\mathrm{NME} = \sum_{i=1}^{n}|y_i - x_i| \Big/ \sum_{i=1}^{n} y_i \tag{10}$$

where $\bar{y}$ is the mean observed value.

4.1. Missing-value imputation

The data loss is incurred by malfunction of sensors, servers, and network transmissions. The

significance of its impact on calibration performance depends on the data loss ratio (DLR) in the training set and the robustness of the missing-value imputation methods. Our collected $PM_{2.5}$ data are recorded in time unit of hours, so there should be 720 (24×30) records for each site in both training and test sets if no value is missing. We check our dataset and calculate the DLR and the maximum time window value is continuously missing in the records. Table 3 tabulates the DLR and the maximum time window (in hours) of the missing value in both training set and test set. We observe that the DLR ranges from 5.8% to 9.2% in the training set, and it is between 1.4% and 3.9% for the test set. The maximum time window of the missing value is 8 or 16 hours for the training set and it is 9 hours for the test set. It is noted that the imputation method is applied only on training data. The test data remains unaltered as being used as the exact values for conducting performance evaluations on the calibrated data. The $PM_{2.5}$ concentration has high variability over time but manifests a daily periodic pattern. The observed $PM_{2.5}$ concentration in a day usually reaches its lowest value around noon and starts climbing in the evening until it reaches its highest value in midnight. The maximum time window of the missing value is 8 or 16 hours in our training set, the daily pattern can be learned and used to replace the missing value. The imputation methods that we applied in the ensemble worked well on our training set as it can be seen in the experimental results. If the maximum time window of the missing value exceeds 24 hours, we suggest the reader to directly remove the null records in order to avoid any bias.

Table 3. Data loss ratio in the monitored data of the low-cost sensors.

	Training set		Test set	
Low-cost sensors	DLR	Max. time window (hours)	DLR	Max. time window (hours)
A1	5.8%	8	3.9%	9
A2	9.2%	8	3.5%	9
A3	7.5%	16	1.4%	9

We apply our ensemble learning approach with or without missing-value imputation learning to obtain the calibration results. To realize the influence of training-data imputation on the test-data calibration, we compute the R^2, RMSE, and NME between the $PM_{2.5}$ readings of the paired supersites and low-cost sensors. The numeral performance is shown in Table 4. It is seen that the R^2, RMSE, and NME between the original readings without performing any imputation and calibration learning is 73.57%, 13.20, and 0.6188, respectively. By performing the proposed ensemble learning on both imputation and calibration, all three optimizers, Sobol, N&M, and PSO, significantly enhance the three performance measures. The improvement ratio to the original measures is listed in parentheses. PSO is the best optimizer which improves the three performance measures by 4.92%, 52.96, and 56.85%, respectively. Sobol is the second best optimizer followed by N&M. Next, we apply the ensemble learning on calibration, however, no imputation is performed on the training set. The three optimizers are still able to perform calibration reasonably well. The improvement ratio decreases by about 1% as compared to performing ensemble learning on both imputation and calibration. The reason may be due to the fact that the DLR in the training set is not too high (see Table 3), so the execution of imputation does not significantly affect the regression trees learned by XGBoost. The ensemble calibration also works best with the PSO optimizer, followed by Sobol and N&M.

Table 4. Calibration performance of low-cost sensors with or without imputation.

	R^2	RMSE	NME
No Imputation and No Calibration:	73.57%	13.20	0.6188
Ensemble Learning on Both Imputation and Calibration:			
Sobol	76.51% (3.99%)	6.27 (52.47%)	0.2677 (56.74%)
N&M	75.25% (2.28%)	6.56 (50.27%)	0.2741 (55.69%)
PSO	77.19% (4.92%)	6.21 (52.96%)	0.2670 (56.85%)
No Imputation, Only Ensemble Calibration:			
Sobol	75.96% (3.25%)	6.32 (52.13%)	0.2708 (56.24%)
N&M	74.69% (1.52%)	6.73 (48.99%)	0.2802 (54.72%)
PSO	76.36% (3.78%)	6.29 (52.35%)	0.2714 (56.14%)

4.2. Calibration

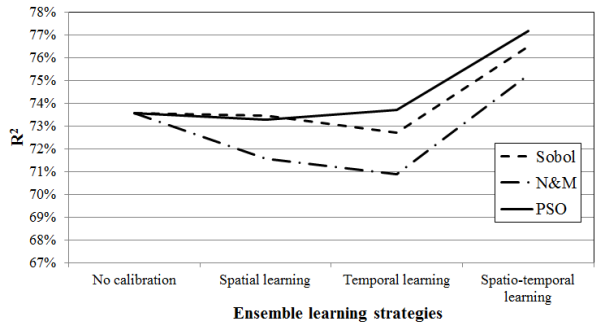
Our ensemble approach learns the best strategy for combining the imputation method, XGBoost hyperparameter values, and spatiotemporal data composition for calibration of low-cost PM2.5 sensors. Our previous experimental result has shown the execution of data imputation does improve the performance of subsequent calibration, however, the influence is not significant. Now, we want to measure the contribution of ensemble learning on spatiotemporal data for calibration. We classify our ensemble learning into three types: ensemble learning on spatial data, ensemble learning on temporal data, and ensemble learning on spatiotemporal data. Each type of ensemble learning is separately applied with each of the three optimizer, namely, Sobol, N&M, and PSO. We detail the comparative performance of various ensemble learning for calibration methods in Table 5, the best calibration result for each low-cost sensor by every type of ensemble learning is shown in boldface under the corresponding performance metric. The baseline result is obtained without applying ensemble learning, i.e., the performance is measured between original test data of the supersites and low-cost sensors. The experimental results have the following implications. (1) It is seen that PSO is the best optimizer for ensemble learning with temporal and spatiotemporal data. PSO also performs well for ensemble learning with spatial data, though Sobol performs slightly better than PSO. N&M is outperformed by the other two optimizer for all types of ensemble learning. (2) All the three types of ensemble learning for calibration can achieve significant improvement on RMSE and NME, but only the ensemble learning on spatiotemporal data is able to enhance the R^2 measure. This phenomenon indicates that the ensemble learning can actively choose the best composition of training set from either spatial or temporal perspectives, and the application of spatiotemporal learning can achieve the overall best calibration performance. (3) Figure 3 shows the mean performances over the three low-lost sensors by using various types of ensemble learning strategies. It is clearly seen that both Sobol and N&M enhance the mean RMSE and NME while slightly deteriorate R^2 at the same time when applying either ensemble spatial learning or ensemble temporal learning. On contrary, PSO makes good explorations in spatial and temporal spaces considering the tradeoffs among the three, sometimes conflicting, performance objectives. Therefore, PSO well surpasses Sobol and N&M with every type of ensemble learning.

Table 5. Calibration performance of microsite sensors with various ensembles.

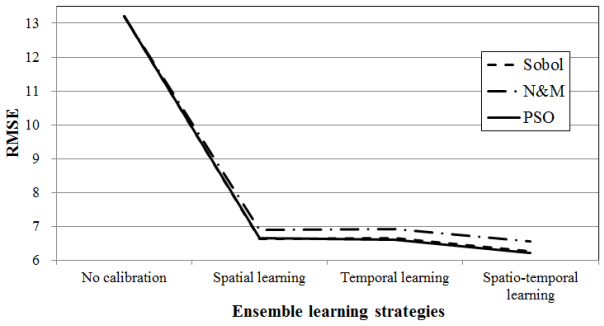
Calibration methods	R^2	RMSE	NME
No Ensemble Learning:			
A1	72.95%	12.95	0.5768
A2	65.55%	15.81	0.8877
A3	82.21%	10.83	0.3918
mean	73.57%	13.20	0.6188
Sobol Ensemble Learning on Spatial Data:			
A1	76.44%	**7.20**	**0.3067**
A2	**64.37%**	**6.99**	0.3592
A3	79.62%	**5.69**	**0.1732**
mean	73.48%	6.63	0.2797
N&M Ensemble Learning on Spatial Data:			
A1	75.34%	7.45	0.3114
A2	60.22%	7.38	0.3670
A3	79.11%	5.84	0.1803
mean	71.56%	6.89	0.2862
PSO Ensemble Learning on Spatial Data:			
A1	**76.48%**	**7.20**	**0.3067**
A2	63.59%	7.04	**0.3583**
A3	**79.73%**	5.71	0.1748
mean	73.27%	6.65	0.2799
Sobol Ensemble Learning on Temporal Data:			
A1	75.60%	**7.19**	0.3087
A2	62.88%	7.02	0.3569
A3	**79.66%**	5.75	0.1748
mean	72.71%	6.65	0.2801
N&M Ensemble Learning on Temporal Data:			
A1	74.68%	7.52	0.3121
A2	58.83%	7.44	0.3659
A3	79.17%	5.83	0.1794
mean	70.89%	6.93	0.2858
PSO Ensemble Learning on Temporal Data:			
A1	**76.26%**	7.20	**0.3062**
A2	**65.46%**	**6.90**	**0.3536**
A3	79.36%	**5.72**	**0.1733**
mean	73.69%	6.61	0.2777
Sobol Ensemble Learning on Spatiotemporal Data:			
A1	**77.43%**	**6.82**	**0.2951**
A2	68.21%	6.96	0.3569
A3	83.88%	5.04	0.1512
mean	76.51%	6.27	0.2677

Continued on next page

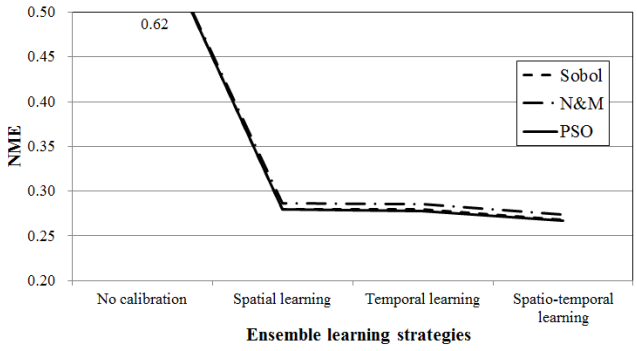
Calibration methods	R2	RMSE	NME
N&M Ensemble Learning on Spatiotemporal Data:			
A1	76.98%	6.84	0.2956
A2	68.78%	7.33	0.3641
A3	80.01%	5.52	0.1628
mean	75.26%	6.56	0.2742
PSO Ensemble Learning on Spatiotemporal Data:			
A1	**77.43%**	6.85	0.2972
A2	**69.98%**	**6.85**	**0.3536**
A3	**84.16%**	**4.93**	**0.1502**
mean	77.19%	6.21	0.2670



(a)



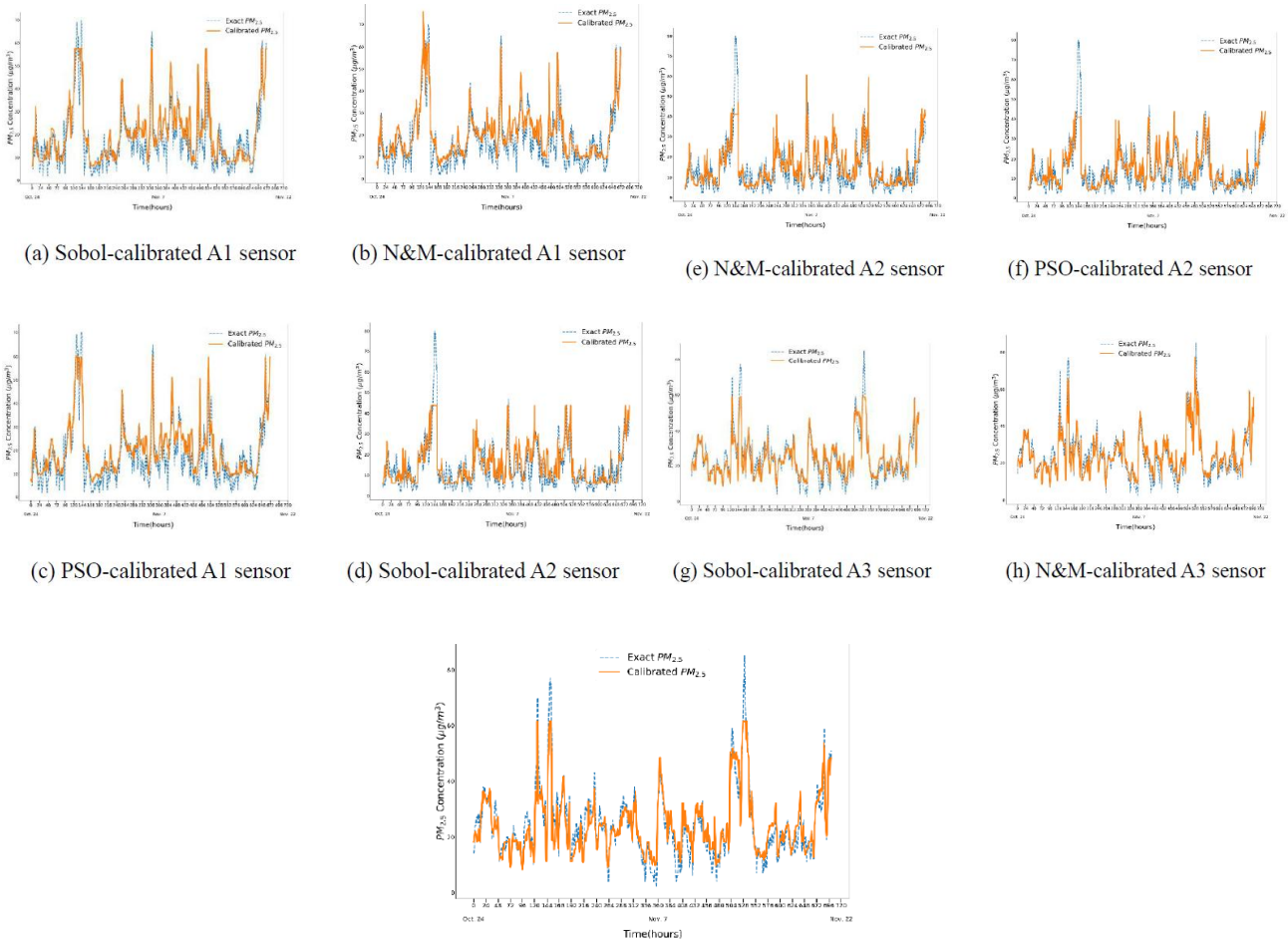
(b)



(c)

Figure 3. Mean performances over the three low-lost sensors by using various ensembles.

Figure 4 shows the hourly comparison between the exact $PM_{2.5}$ (readings of the supersite) and the calibrated $PM_{2.5}$ (calibrated readings of the low-cost sensor) for the three sites. For each site, we separately show the calibrated curve obtained by Sobol, N&M, and PSO, respectively. All the three optimizers apply ensemble hyperparameters and spatiotemporal data learning to automatically calibrate the readings from the low-cost site to align with the exact readings reported from the expensive government-built supersite. But N&M seems to over-calibrate the readings at some epochs and manifest many fluctuations in the calibration, this behaviour is particularly conspicuous as can be seen in Figure 4(e). On contrary, Sobol and PSO tend to be relatively conservative in calibrating peak and valley values as compared to N&M. One promising direction of our future research is to blend the calibration results from the three optimizers to adaptively capture the main trend and fluctuating details of the exact $PM_{2.5}$ series.



(a) Sobol-calibrated A1 sensor (b) N&M-calibrated A1 sensor (e) N&M-calibrated A2 sensor (f) PSO-calibrated A2 sensor

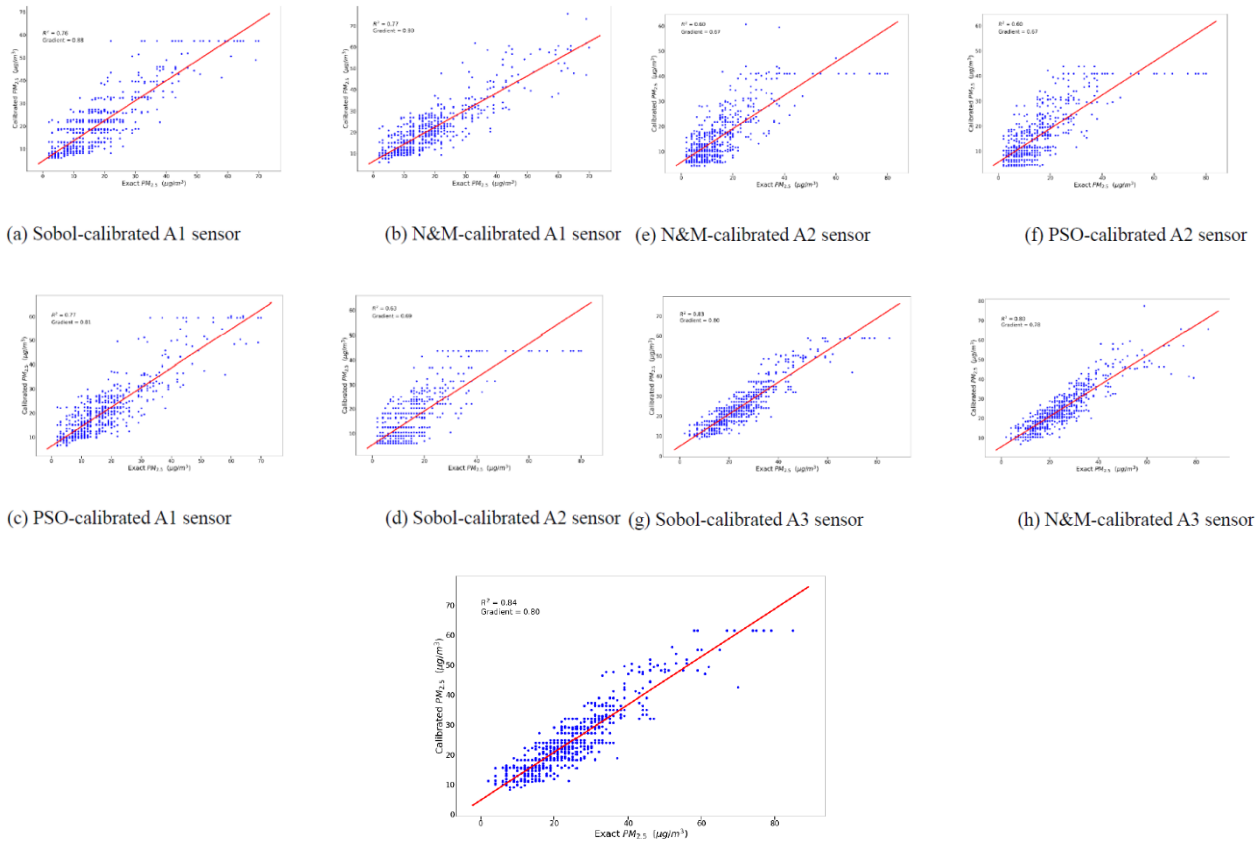
(c) PSO-calibrated A1 sensor (d) Sobol-calibrated A2 sensor (g) Sobol-calibrated A3 sensor (h) N&M-calibrated A3 sensor

(i) PSO-calibrated A3 sensor

Figure 4. Hourly comparison between the exact $PM_{2.5}$ and the calibrated $PM_{2.5}$ obtained by Sobol, N&M, and PSO.

Figure 5 shows the scatter plots and gradient between the exact $PM_{2.5}$ and the calibrated $PM_{2.5}$ obtained by Sobol, N&M, and PSO. The gradient ranges from 0.67 to 0.88. The highest gradient is obtained by applying Sobol to calibrate A1 sensor, and the lowest gradient is observed when A2 sensor is calibrated by N&M or PSO. The gradient so obtained is mainly influenced by the selected

supersite rather than the applied optimization method. A2 sensor tends to produce a lower gradient than those obtained with A1 sensor and A3 sensor. The comparative results of gradient conform to those of other performance metrics observed in Table 5.



(a) Sobol-calibrated A1 sensor (b) N&M-calibrated A1 sensor (e) N&M-calibrated A2 sensor (f) PSO-calibrated A2 sensor

(c) PSO-calibrated A1 sensor (d) Sobol-calibrated A2 sensor (g) Sobol-calibrated A3 sensor (h) N&M-calibrated A3 sensor

(i) PSO-calibrated A3 sensor

Figure 5. Scatter plots and gradient between the exact $PM_{2.5}$ and the calibrated $PM_{2.5}$ obtained by Sobol, N&M, and PSO.

5. Conclusions

Establishing a low-cost $PM_{2.5}$ sensor IoT is complement to the AQI monitoring network of government-built supersites. Low-cost sensors provide a dense coverage of monitoring area but lack high-accuracy measurements. Calibration of low-cost sensor measurements by reference to high-accuracy supersites is cheap and automatic. In this paper, we have proposed a novel ensemble approach for learning the best strategy to select the imputation method, hyperparameter values of calibration model, and the composition of spatiotemporal data. Three optimizers, namely, Sobol, N&M, and PSO, were tested with various kinds of ensemble learning. The experimental results show that our ensemble method actively learns the optimal strategy for combining imputation, parameterization of calibration, and composition of spatiotemporal data. The best performance is obtained by using PSO, and the improvement ratio with respect to R^2, RMSE, and NME, is 4.92%, 52.96%, and 56.85%, respectively.

Acknowledgments

This research is partially supported by Ministry of Science and Technology of ROC, under Grant MOST 107-2410-H-260 -015 -MY3, Grant MOST 107-2420-H-260-002-HS3, and Environmental Protection Administration of ROC, under Grant EPA-107-FA12-03-A150.

Conflict of interest

We declare no conflicts of interest in this paper.

References

1. C. Song, J. He, L. Wu, et al., Health burden attributable to ambient $PM_{2.5}$ in China, *Environ. Pollut.*, **223** (2017), 575–586.
2. IPCC, Climate Change 2007: the Scientific Basis, Contribution of Working Group I, in *T*hird Assessment Report of the Intergovernmental Panel on Climate Change (eds. J. T. Houghton, Y. Ding, D. J. Griggs, et al.), Cambridge University, New York (2007).
3. Y. J. Liu, T. T. Zhang, Q. Y. Liu, et al., Seasonal variation of physical and chemical properties in TSP, PM10 and $PM_{2.5}$ at a roadside site in Beijing and their influence on atmospheric visibility, *Aerosol Air Qual. Res.*, **14** (2014), 954–969.
4. L. Mo, Z. Ma, Y. Xu, et al., Assessing the capacity of plant species to accumulate particulate matter in Beijing, China, *PLoS One*, **10** (2015), 0140664.
5. K. Hu, A. Rahman, H. Bhrugubanda, et al., HazeEst machine learning based metropolitan air pollution estimation from fixed and mobile sensors, *IEEE Sens. J.*, **17** (2017), 3517–3525.
6. M. Miksys, Predictions of $PM_{2.5}$ and PM10 concentrations using static and mobile sensors, Technical Report, School of Informatics, University of Edinburgh, (2016).
7. L. J. Chen, Y. H. Ho, H. C. Lee, et al., An open framework for participatory $PM_{2.5}$ monitoring in smart cities, *IEEE Access*, **5** (2017), 14441–14454.
8. S. Ausati and J. Amanollahi, Assessing the accuracy of ANFIS, EEMD-GRNN, PCR, and MLR models in predicting $PM_{2.5}$, *Atmos. Environ.*, **142** (2016), 465–474.
9. H. W. Barker, Isolating the industrial contribution of $PM_{2.5}$ in Hamilton and Burlington, *Ontario J. Appl. Meteorol. Climatol.*, **52** (2013), 660–667.
10. M. Jerrett, R.T. Burnett, R. Ma, et al., Spatial analysis of air pollution and mortality in Los Angeles, *Epidemiology*, **16** (2005), 727–736.
11. A. Di Antonio, O. Popoola, B. Ouyang, et al., Developing a relative humidity correction for low-cost sensors measuring ambient particulate matter, *Sensors*, **18** (2018), 2790.
12. G. J. Hwang and S. S. Tseng, A heuristic task-assignment algorithm to maximize reliability of a distributed system, *IEEE T. Reliab.*, **42** (1993), 408–416.
13. I. Eekhout and R. M. de Boer, Missing data: a systematic review of how they are reported and handled, *Epidemiology*, **23** (2012), 729–732.
14. H. Shen, X. Li and Q. Cheng, Missing information reconstruction of remote sensing data: A technical review, *IEEE Geosc. Rem. Sen. M.*, **3** (2015), 61–85.
15. S. A. Dudani, The distance-weighted k-nearest-neighbor rule, *IEEE T. Syst. Man. Cy.*, **4** (1976), 325–327.

16. M. J. Azur, E. A. Stuart, C. Frangakis, et al., Multiple imputation by chained equations: what is it and how does it work? *Int. J. Meth. Psych. Res.*, **20** (2011), 40–49,.
17. R. Mazumder, T. Hastie and R. Tibshirani, Spectral regularization algorithms for learning large incomplete matrices, *J. Mach. Learn. Res.*, (2010), 2287–2322.
18. P. F. Pai, K. P. Lin, C. S. Lin, et al., Time series forecasting by a seasonal support vector regression model, *Expert Syst. Appl.*, **37** (2010), 4261–4265.
19. J. Friedman, Greedy function approximation: a gradient boosting machine. *Ann. Stat.*, **29** (2001), 1189–1232.
20. T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2016), San Francisco, USA, (2016).
21. R. Bekkerman, The present and the future of the KDD cup competition: an outsider's perspective. Available from: https://www.linkedin.com/pulse/present-future-kdd-cup-competition-outsiders-ron-bekkerman. Commentary from Linkedin at on Aug. 25, 2015.
22. J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, *J. Mach. Learn. Res.*, (2012), 281–305.
23. H. Wang, Q. Geng and Z. Qiao, Parameter tuning of particle swarm optimization by using Taguchi method and its application to motor design, in Proceedings of the 4th IEEE International Conference on Information Science and Technology, (2014).
24. J. Safarik, J. Jalowiczor, E. Gresak, et al., Genetic algorithm for automatic tuning of neural network hyperparameters, in Proceedings of SPIE Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything, (2018).
25. J. Derrac, S. Garc á and F. Herrera, A Survey on Evolutionary Instance Selection and Generation, *Int. J. Appl. Metaheuristic Comput.*, **1** (2010), 60–92.
26. J. Kennedy and R. C. Eberhart, Particle swarm optimization, in Proceedings IEEE International Conference on Neural Networks IV, (1995), 1942–1948.
27. S. Joe and F. Y. Kuo, Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator, *ACM T. Math. Software*, **1** (2003), 49–57.
28. J. C. Lagarias, J. A. Reeds, M. H. Wright, et al., Convergence properties of the Nelder-Mead simplex method in low dimensions, *SIAM J. Optimiz.*, **9** (1998), 112–147.
29. B. K. Tan, Laboratory evaluation of low to medium cost particle sensors, Master's Thesis, University of Waterloo (2017).
30. K. E. Kelly, J. Whitaker, A. Petty, et al., Ambient and laboratory evaluation of a low-cost particulate matter sensor, *Environ. Pollut.*, **221** (2017), 491–500.

AIMS Press